DATE 23/8/23 (46)

Software Development Life Cycle

- SDLC is a framework that describes the activities performed at each stage of the software development project

- ~~The aim of~~ SDLC is to design, develop & ~~test~~ high quality software by ~~the~~ processes that are used by ~~the~~ software industry.

⇒ AIM

- SDLC is a process used by software industry to design, develop and test high quality softwares.

⇒ PHASES → includes budgeting.

1. Planning & Requirement Gathering & Analysis
2. Design ~~of~~ the Roadmap
3. Development
4. Testing
5. Implementation
6. Maintainance

⇒ PHASE I [Planning, Req. Gathering & Analysis]

- Phase I defines the importance of system development lifecycle (SDLC)
- Also called SDLC planning phase.
- Includes various processes like market research, conduct customer interviews, research ~~on~~ your competition & conduct surveys, etc.
- The feedback that is gathered can be used to analyze product-market fit
- Phase I also defines what needs to be in the product and what doesn't needs to be in product

⇒ PHASE II [SDLC Designing Phase]
- This phase II relates to the design of the product.
- It is essential to do as much visualization as possible about the product in this phase
- Design phase is majorly used to communicate the plans for the product to all the developers.
- Visualization can include 3d modellings, cardboard/plastic modellings, framework designing, etc.

⇒ PHASE III [Coding/Programming/Development Phase]
- Coding phase in SDLC is not handled directly by Product Managers.
- It is handled by the Software/program developers.
- The development phase is crucial as the product is built ultimately in this phase

⇒ PHASE IV [Testing Phase]
- SDLC models all hinge on the testing phase
- This phase is essential to ensure all the features of the product are working functional
- Models → AGILE ⇒ Frequent testing
  → WATERFALL ⇒ Not-frequently tested.

⇒ PHASE V [DEployment]
- In Development phase, the application is made available to the users.
- Many companies prefer to automate the deployment phase.

- It can be as simple as a payment portal and download link on the company's website

⇒ 6 PHASE 6 [Operations & Maintainance]
- At this point, deployment cycle is almost complete & the product is being used in the field.
- It is also an important phase, as here the users discover bugs that were unnoticed during testing.
- These bugs/errors needs to be resolved, which can spawn new development cycles.

✱ SDLC Models
- to help to understand & implement the SDLC phases, various SDLC models

→ Reasons for using SDLC Models.
- provides basis for project planning, estimation & scheduling.
- provides framework for standard set of terminologies, activities & deliverables
- provides mechanism for project tracking & control.
- Increases visibility of project progress to all the stakeholders.

⇒ Advantages of selecting appropriate SDLC Model
- increased development speed.
- increased product quality.
- improved tracking & control
- improved client relationship.
- decreased product/project errors/bugs risks.
- decreased project management overhead.
  ↳ distribution of project work.

⇒ Common SDLC Models
- Waterfall
- Spiral / Iterative
- Agile

⇒ SDLC Waterfall Model

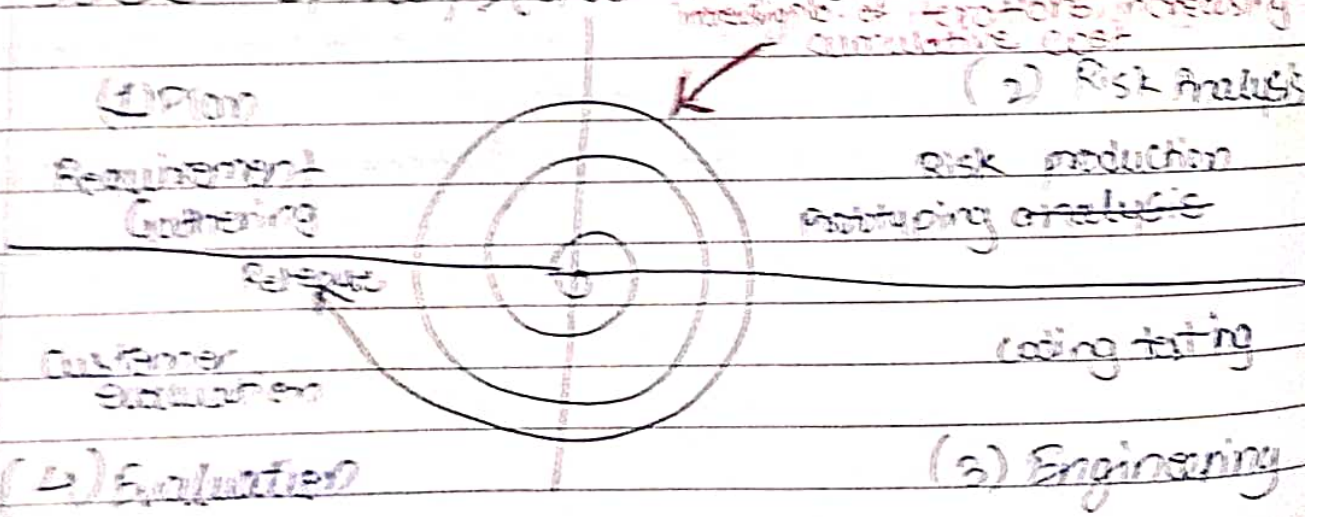
Analysis →
Design →
Development →
Testing →
Deployment →
Maintainance

- If any error encountered, you can't go back to the previous phase & will need to go the starting/analysis phase.
- oldest & most well-known model.
- follows sequential step-by-step process from req. analysis to maintainance.
- used majorly for manufacturing of physical product

→ Strengths of SDLC Waterfall Model
- Easy to understand & use
- Provides structure / framework to inexperienced staff.
- Milestones / Steps / Phases are well-understood.
- Sets requirement stability
- Good for management control (plan, staff, track)
- Works more efficiently & well when quality is more important than cost or schedule.

→ Weaknesses
- All requirements must be fully ~~stated~~ specified upfront.
- Deliverables created for each phase are ~~consi~~ considered frozen → inhibits flexibility
- Can give false impression of progress
- Does not reflect problem-solving nature of software development → iterations of phases
- Little opportunity for customer to preview the system (until it may be too late).

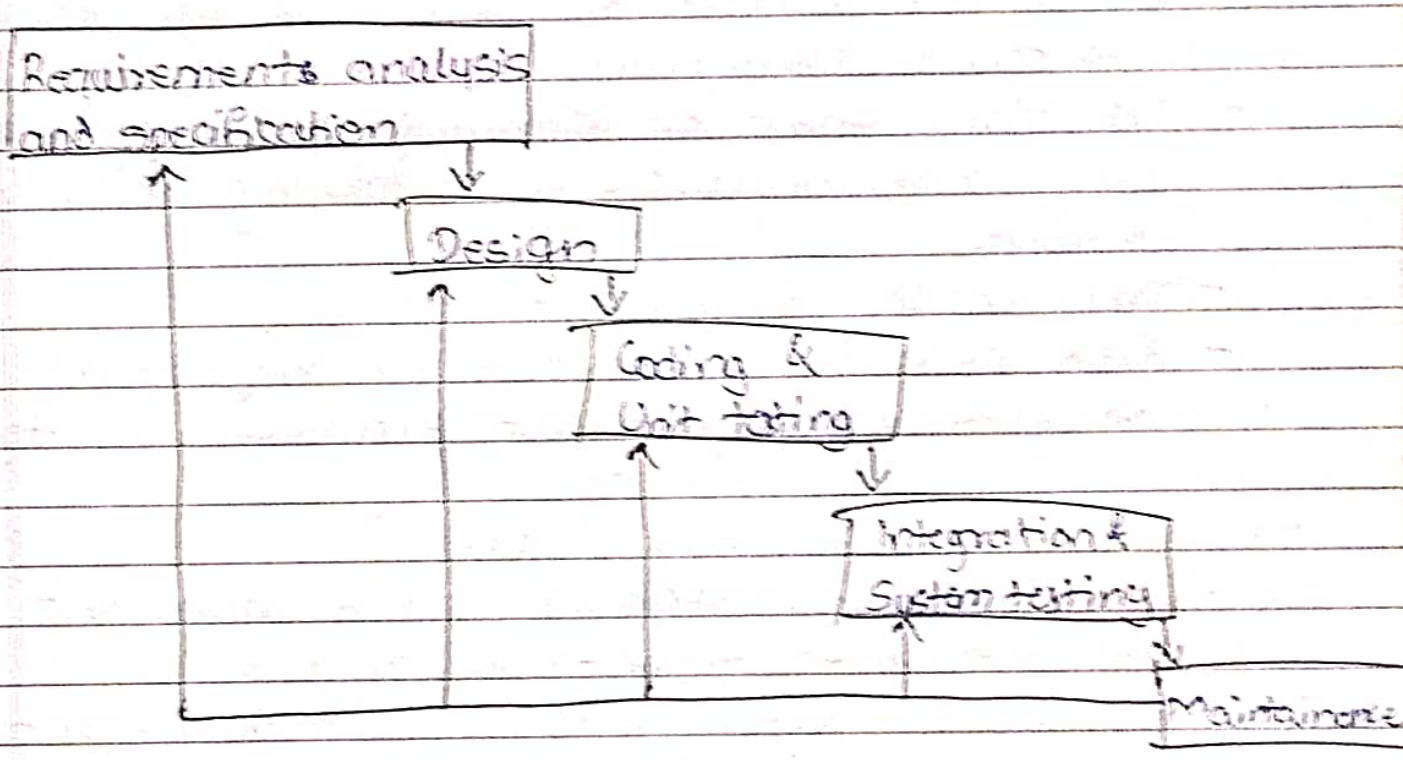
⇒ SDLC Spiral/Iterative Model



Date
SUS 23 (57)

- It is a risk-driven (prioritizes risk) iterative model
- Spiral model is also called risk model or iterative model.
- It will take 6 months to 2 years to complete the project.
- Each iterative starts with small set of requirements and goes through development phase (except installation & maintainance) for those set of requirements

→ After iterative/spiral model it is necessary to do the a bi-directional waterfall model.



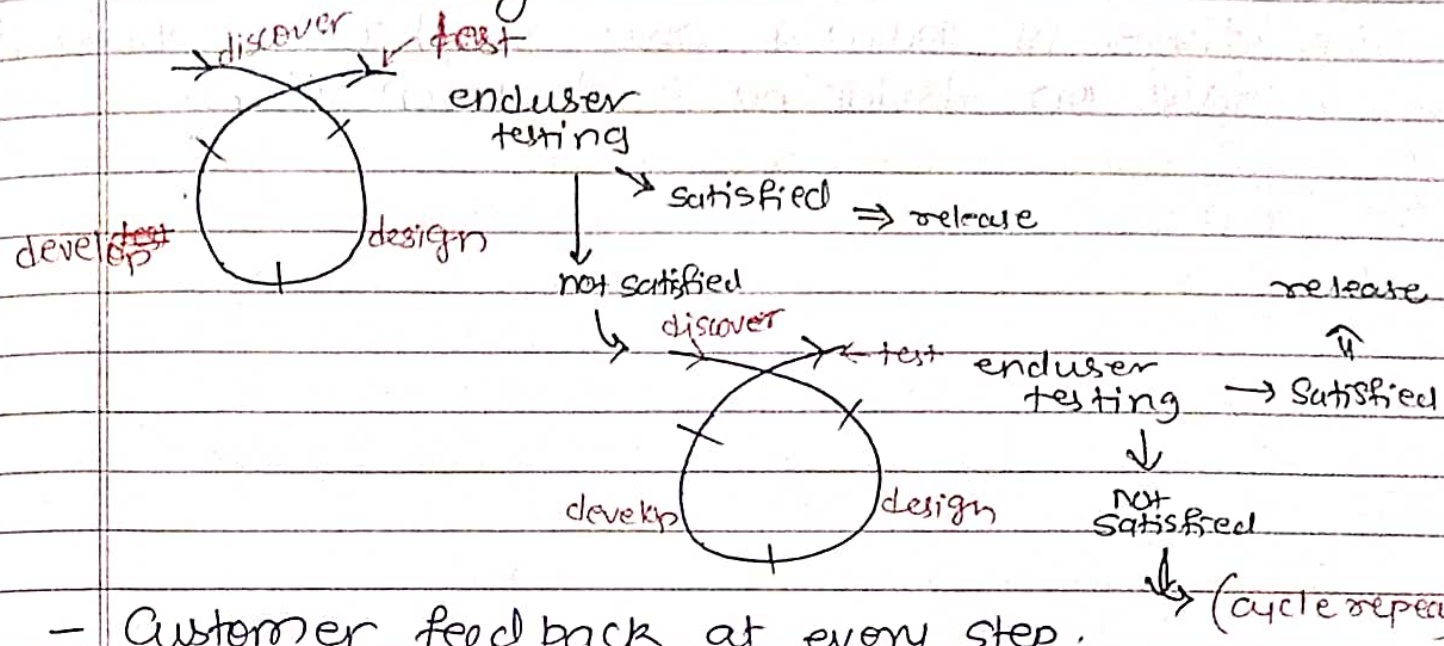
- Iterate or spiral model is followed until all major risks are addressed and application is ready for the installation and maintainance phase (production)
- Last iteration is the waterfall process.

→ Strengths of SDLC Spiral / Iterative Model
- Critical high-risk functions are developed & solved first. the design / product has low chance of beind errored or defective
  ↳ Design doesn't have to be perfect.
- Users can see / use products / system early because of rapid prototyping tools.
- Users can be closely tied to all lifecycle steps to have early & frequent feedbacks. from users.

→ Weaknesses of SDLC Spiral / Iterative Model
- Time spent for evaluating risks is too large even for small or low-risk products / system
  ↳ time consuming
- Also time spent on planning, resetting objectives doing risk analysis & prototyping may be excessive
- The model is complex
- Risk assessment expertise is required.
  ↳ increasing project cost / Expensive to an extent.

→ When to use Spiral Model?
- when risk-evaluation is more important
- for medium-to-high-risk projects.
- the product / system is to be built from scratch and isn't available in the market.
  ↳ requirements becomes unclear & complex
    ↳ users / developers are unsure of the project nee
- creating a new product line.

⇒ # SDLC Agile Model



- Customer feedback at every step.
- used for time-critical applications
- The time frame of each iteration in Agile process model is short and typically lasts from one to four weeks.

→ Strengths of Agile Model

- Delivery of the proper working product faster than conventional linear development model.
- Customer feedback at every step/stage ensures that the end ~~deliver~~ deliverable product satisfies the client's expectations.
- No guesswork between the development team and the customer, as there is ~~no~~ face to face communication & continuous inputs from the clients.

→ Weaknesses of Agile Model.

- For longer projects, it is difficult to judge the efforts and time required for the project in SDLC.
- It is difficult to design product as the requirements are constantly changing due to client's inputs.

- Since the requirements are ever changing, there is hardly any emphasis, which is laid on designing & documentation.